A Pulse Width Modulation based Power-elastic and Robust Mixed-signal Perceptron Design

Sergey Mileiko†, Rishad Shafik†, Alex Yakovlev†, Jonathan Edwards‡
†Newcastle University, UK; ‡Temporal Computing, UK

***Abstract*—Neural networks are exerting burgeoning influence in emerging artificial intelligence applications at the micro-edge, such as sensing systems and image processing. As many of these systems are typically self-powered, their circuits are expected to be resilient and efficient in the presence of continuous power variations caused by the harvesters. In this paper, we propose a novel mixed-signal (i.e. analogue/digital) approach of designing a power-elastic perceptron using the principle of pulse width modulation (PWM). Fundamental to the design are a number of parallel inverters that transcode the input-weight pairs based on the principle of PWM duty cycle. Since PWM-based inverters are typically agnostic to amplitude and frequency variations, the perceptron shows a high degree of power elasticity and robustness under these variations. We show extensive design analysis in Cadence Analog Design Environment tool using a 3 x 3 perceptron circuit as a case study to demonstrate the resilience in the presence of parameric variations.**

I. Introduction and Motivation

Perceptron is the basic building block of deep neural networks used in machine learning applications [1] [2] [3]. It consists of an input vector, a set of weights and a bias to produce binary classification outcomes, as follows:

$$f(x) = \begin{cases} 1, & \text{if } \mathbf{w}.\mathbf{x} + b > 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where w is a vector of real-valued weights, **w**.**x** is the dot product $\sum_{i=1}^{m} w_i x_i$ with m number of inputs, and b is the bias. The process of deciding the appropriate weights (**w**), often also known as training, serves as the basic principle of supervised learning. When m becomes large, it approximates the behaviour of a biological neuron.

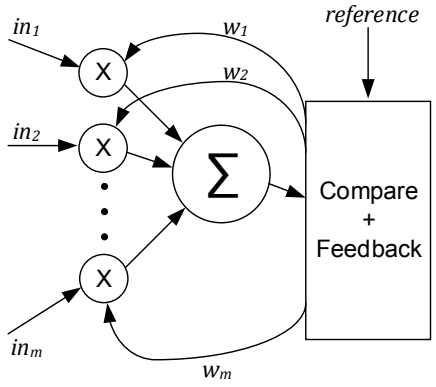

Fig. 1. The structure of perceptron.

Fig. 1 shows the typical structure of a perceptron [4] [5]. The core of it is an adder that adds m weighted inputs. The result of the addition is compared with a reference during the training phase. During this time, the weights are updated to ensure the reference is matched.

For hardware implementation multiplication and addition are crucial arithmetic circuits in a perceptron [6]. Such arithmetic operations require significant area and power costs, which depend on the number of input-weight pairs and the precision of the multipliers and adders.

Future micro-edge applications will be increasingly autonomous. The key for autonomy is being not only based on smartness through machine learning capability but also being able to work from energy-harvesting sources. In other words, the circuits must be capable of working under a dynamic range of power variation.

For a class of applications involving machine learning at the micro-edge, such as sensors with data filtering and compression, we can envisage power to be extracted from the environment [7]. Energy-harvesting power sources may not be always equipped with accurate power regulation circuits, which are themselves power-hungry. Hence, in this work we are aiming at developing a perceptron design that is resilient to power variations, i.e. power-elastic [8].

Existing perceptron designs are predominantly digital; although a number of analogue implementations have been reported [9] [10]. Nonetheless, these designs are vulnerable to power supply variations. As such, these are not suitable for working under extreme power variations. In other words, these have poor power elasticity properties that refrain them from providing useful computation under unreliable power supplies.

To address power elasticity, which gives reliable results even with unstable supply voltage and input frequency we propose to transfer the arithmetical computation process from the digital domain to the temporal domain, where the information is encoded in the input pulse width. This would guarantee the reliability of the input data, because the pulse width (input signal duty cycle) is not affected neither by the input frequency, nor by its amplitude.

The aim of this paper is to design a power and frequency elastic perceptron, which performs the arithmetic computation in the PWM-coded format. The main ***contributions*** are:

1) a mixed-signal perceptron design using duty cycle based temporal weight encoding and input switching via a PWM inverter, and
2) extensive validation experiments in Cadence Analog Design tool demonstrating its resilience in the presence of amplitude and frequency variations.

The rest of the paper is organized as follows. Section II presents the proposed design approach. Section III validates the approach using a number of parametric sweeps to demonstrate power elasticity and resilience. Finally, Section IV discusses and concludes the paper.

II. Proposed Approach

The proposed approach is based on the principle that if the input of an inverter is a periodic signal, such as clock, the average voltage on its output is inversely proportional to the duty cycle of the input clock. This is due to the fact that during the interval of time when the input is Low the output capacitance is charged with current from the power source via the PMOS transistor, and during the interval of input being High the capacitance is discharged via the NMOS transistor. Since an inverter is a digital component, whose output equals to logic $'0'$ or $'1'$, it should be "analogized" (i.e. transcoded) in order to convert the input duty cycle into the output voltage that is a corresponding proportion of the supply voltage. This may be achieved by the following ways:

- increasing the input switching frequency,
- increasing the output capacitance, and
- limiting the output current.

Fig. 2 shows the inverter circuit that meets the requirements. The output capacitance of the inverter has been increased by adding a capacitor C_{out} between the output of the inverter and ground. The output resistor R_{out} performs several functions. Firstly, it limits the current, increasing the capacitor's charging/ discharging time. Secondly, it reduces the system's power consumption (See Section III). And, lastly, it adds linearity to the output characteristics as the PMOS and NMOS resistance may be different with different drain voltages. A large resistive load can neglect this difference, the demonstration of which will be shown in Section III.

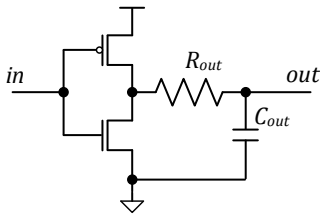

Fig. 2. An inverter with the output resistor and capacitor.

A key feature of this circuit is that if we connect the outputs of several cells, the resulting output voltage will be inversely proportional to the average value of the inputs duty cycle. Therefore, using these inverters, we can build an adder with the PWM-coded inputs, leading to analog output.

To design a perceptron the ability to integrate weighted adders is another crucial design requirement. The adders must be capable of programming the input weights, when required. This is performed by replacing the inverters by AND gates. One input of this gate is the PWM-coded, and another is a digital switch for enabling or disabling this cell. Fig. 3 shows a perceptron architecture with 3×3 weighted adder, built with such gates.

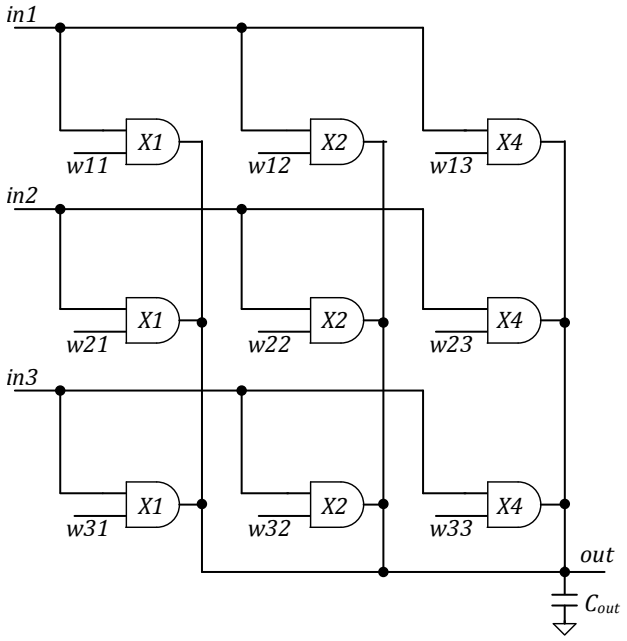

Fig. 3. 3x3 weighted adder.

As can be seen, the circuit adds 3 PWM-coded inputs multiplied by 3-bit weights. Every weight bit is implemented on a separate cell. The least significant bit goes to the cell with the smallest transistor size and the largest output resistor (cells $X1$). The second bit is computed at the cell with doubled transistors width and halved output resistance (cells $X2$). And the most significant bit is coded with 4 times wider transistors, and 4 times smaller output resistor (cells $X4$).

The output voltage of this adder is calculated as follows:

$$V_{out} = (V_{dd} - GND) \cdot \frac{\sum_{i=1}^{k} DC_i \cdot W_i}{k \cdot (2^n - 1)}. \qquad (2)$$

where k is the number of the inputs, n is the number of bits of the weight, DC_i is the duty cycle of the input i, and W_i is the weight of the input i.

The transcoding of spatial data (in digital form) to temporal domain (in PWM duty cycle), and the mixed-mode (analogue/ digital) multiplier/adder operation have significant impact on the circuit complexity, and its resilience in the presence of amplitude and frequency variations. These will be extensively validated in the next section.

III. Experimental Results

A prototype circuit (based on Fig. 2 and 3) is designed using UMC65nm technology and simulated in the Cadence Analog Design Environment tool.

The ability of an inverter to convert a PWM-coded signal into analog is demonstrated by the following experiment. The circuit from Fig. 2 has been simulated with the parameters listed in Table I. These parameters have been optimized after extensive sweep experiments. For brevity, these optimization experiments are not reported here.

TABLE I
SIMULATION PARAMETERS USED IN EXPERIMENTS

Input signal frequency	$V_{dd} = 2.5V$
Transistors width	$n_{width} = 320nm$, $p_{width} = 865nm$
Transistors length	$n_{length} = p_{length} = 1.2um$
Output capacitor	$C_{out} = 1pF$

Fig. 4 shows the dependency of the output voltage from the input signal duty cycle for different sizes of the output load resistor R_{out}.

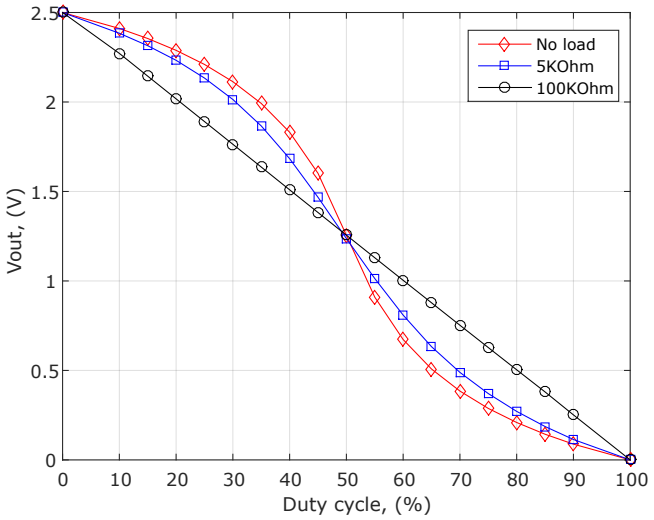

Fig. 4. Output voltage vs input duty cycle of the inverter cell.

The plot shows that the output voltage is reversely proportional to the input signal duty cycle. However, this proportionality is not linear for the inverter with small load resistor and without it. This is caused by the non-linearity of the transistors. Their resistance depends on their drain-to-source voltages, and can be different with different V_{out}. In the case of the large output resistor, it brings the greatest contribution to the overall resistance, and the output function becomes purely linear.

Fig. 5 demonstrates the resilience of the cell to the input frequency variation. Parameters of the simulation are the same an in the previous one (see Table I). The circuit uses fixed value of the output resistor $R_{out} = 100KOhm$. The plot shows the output voltage for the input frequencies from 1MHz to 1500MHz, and input duty cycles 25%, 50%, and 75%. As can be seen, the values of V_{out} are almost the same for a wide range of frequencies. This demonstrates a high degree of frequency resilience for the proposed perceptron design.

To demonstrate the perceptron resilience to the power variations we simulated the inverter circuit with different values of the supply voltage and input amplitude. The results are shown in Fig. 6. The simulations parameters are the same as shown in Table I. The input signal frequency f_{in} is constant and equals to 500MHz.

As can be seen, the output voltage grows almost linearly with increased V_{dd}. As expected, higher duty cycle shows lower output voltage trends, or vice versa. In the case of the unstable supply voltage, the absolute value of the output voltage does not bear any reliable information. In this case, we

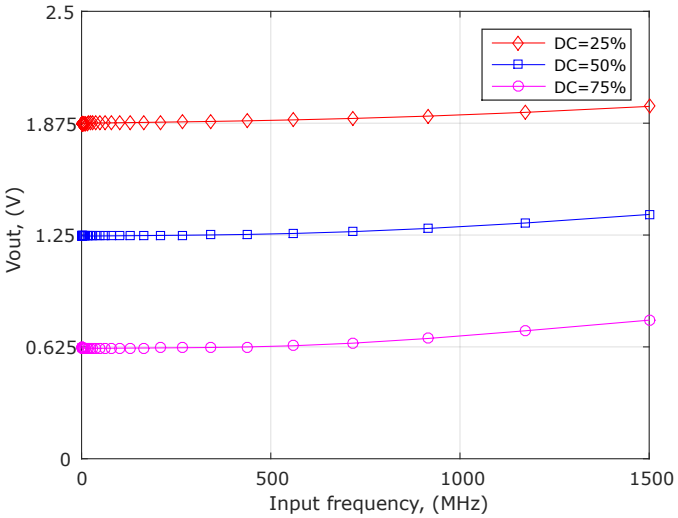

Fig. 5. Output voltage vs input frequency of the inverter cell.

should consider the relation between the output voltage and the supply voltage. This relation will be proportional to the input duty cycle independently from the V_{dd}. This is demonstrated by Fig. 7 where the y axis represents not the absolute value of V_{out}, but the relation of V_{out} to V_{dd} that is more relevant for unstable power conditions.

The circuit demonstrates high resilience to the supply voltage variations. Starting from 1 - 1.5V the relationship of the V_{out} to V_{dd} remains the same for different duty cycles of the input signal.

The simulations below demonstrate the correctness of operation of the 3×3 weighted adder shown in Fig. 3. The parameters of the simulations are the same as in previous, except the size of the output capacitor that has been extended to 10pF. In this simulation we set up different values of three inputs (their weight and duty cycle), and compared the resulted output voltage with its theoretical values (calculated using the formula 2). The results are shown in Table II.

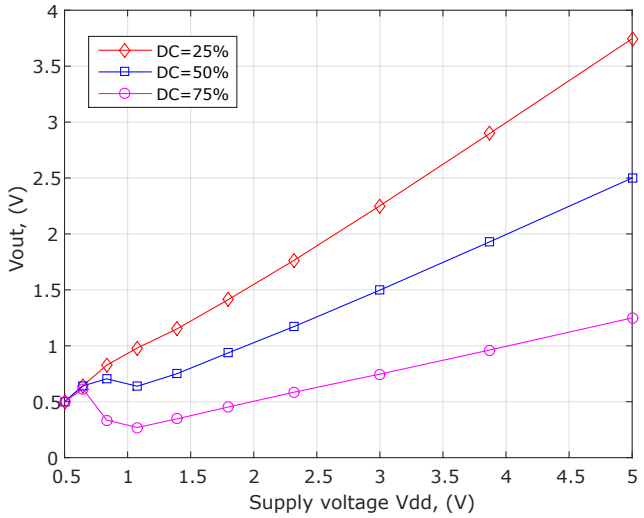

Fig. 6. Output voltage (absolute value) vs power supply.

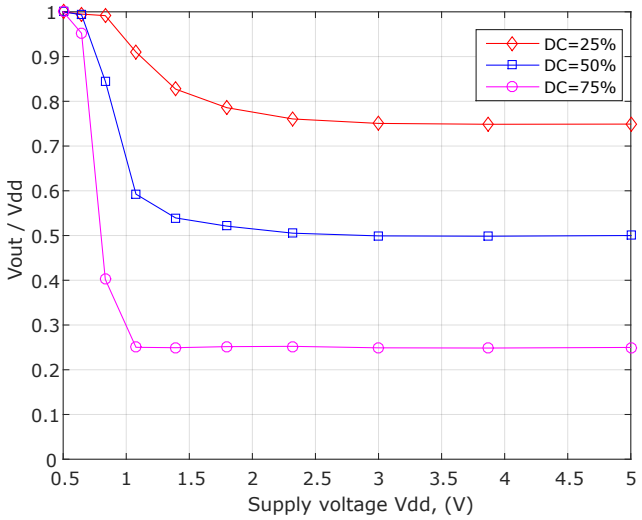

Fig. 7. Output voltage (relative to the power supply) vs power supply.

TABLE II
THE RESULTS OF THE 3×3 WEIGHTED ADDER.

DC1	W1	DC2	W2	DC3	W3	V_{out} theoretical	V_{out} simulation
70%	7	80%	7	90%	7	2.00V	1.99V
50%	1	50%	2	50%	4	0.42V	0.39V
20%	5	60%	6	80%	7	1.21V	1.17V
95%	7	90%	6	80%	6	2.00V	2.05V
30%	1	40%	4	50%	2	0.34V	0.29V
80%	7	20%	3	50%	4	0.96V	0.89V

The simulations results correspond to the theoretical ones, however, the relative error is quite large, especially for the lower output voltages. Despite of this, such errors are still affordable, especially, in the case of perceptron that is a-priori not accurate.

The simulations have been conducted with various input frequencies in the range from $1MHz$ to $1GHz$, but the frequencies did not have any effect on the results, and are not displayed in the table for brevity.

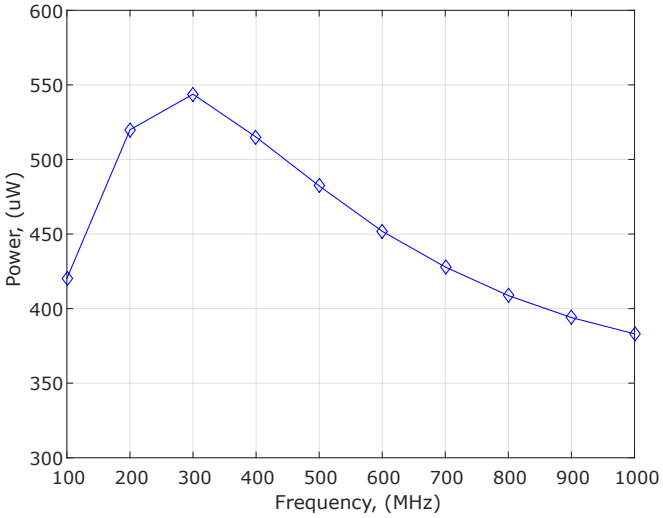

Fig. 8. Average power consumption vs input frequency.

The Fig. 8 shows the power consumption of the designed perceptron for different frequencies. The range of the power may vary within several orders of magnitude depending on the parameters of the perceptron such as sizes of the output resistor and capacitor.

IV. Conclusion and Discussions

We proposed the first mixed-signal (analogue/digital) perceptron design using the principle of PWM. Central to our design are a number of parallel inverters that suitable transcode the input-weight pairs from spatial domain to temporal domain. Since PWM-based inverters are typically agnostic to amplitude and frequency variations, the perceptron shows a high degree of power elasticity and robustness under these variations.

Another advantage of the proposed design is its simplicity. While the conventional implementations of the perceptron require complex logic to perform the multiplication and addition, the proposed approach uses only one gate for per bit for every input. Thus, for the 3×3 weighted adder we used only 54 transistors. This significantly reduces the logic utilization and, thereafter, the power consumption of the entire device.

Machine learning is finding more applications at the micro-edge, where power variation from energy harvesters is becoming commonplace. We believe the proposed perceptron will find practical implementations in these applications as it is highly robust to these variations. This design would nicely complement a power-elastic PWM signal generator based on a self-timed loadable modulo N counter presented in [8].

References

[1] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design*. Boston, MA, USA: PWS Publishing Co., 1996.

[2] E. Wilson and D. W. Tufts, "Multilayer perceptron design algorithm," in *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, 1994, pp. 61–68.

[3] H. Adeli and C. Yeh, "Perceptron learning in engineering design," *Computer-Aided Civil and Infrastructure Engineering*, vol. 4, no. 4, pp. 247–256. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8667.1989.tb00026.x

[4] S. Hung and H. Adeli, "A model of perceptron learning with a hidden layer for engineering design," *Neurocomputing*, vol. 3, no. 1, pp. 3 – 14, 1991. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0925231291900165

[5] B. Jeong and Y. H. Lee, "Design of weighted order statistic filters using the perceptron algorithm," *IEEE Transactions on Signal Processing*, vol. 42, no. 11, pp. 3264–3269, 1994.

[6] W. Qinruo, Y. Bo, X. Yun, and L. Bingru, "The hardware structure design of perceptron with fpga implementation," in *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*, vol. 1, 2003, pp. 762–767 vol.1.

[7] R. Shafik, A. Yakovlev, and S. Das, "Real-power computing," *IEEE Transactions on Computers*, vol. 67, no. 10, pp. 1445–1461, 2018.

[8] O. Benafa, D. Sokolov, and A. Yakovlev, "Loadable Kessels counter," in *Proceedings of ASYNC 2018*, Vienna, May 2018.

[9] R. LiKamWa, Y. Hou, Y. Gao, M. Polansky, and L. Zhong, "Redeye: Analog convnet image sensor architecture for continuous mobile vision," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016, pp. 255–266.

[10] Chen, Yu-Hsin and Krishna, Tushar and Emer, Joel and Sze, Vivienne, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," in *IEEE International Solid-State Circuits Conference, ISSCC 2016, Digest of Technical Papers*, 2016, pp. 262–263.